Midterm 3 Study Guide

The test covers all the material discussed in sections 9.4-9.7, the CFG handout (sections 5.1 and 5.2.1), and the pushdown-automata[1] and parsing[2] notes, and homeworks 9 through 12. The following set of questions is meant to help guide your study and is not meant to be exhaustive of all the possibilities.

1. (review from midterm 2) Suppose that problem A Turing reduces to problem B. Which of the following are necessarily true?

    - If A is computable then B is computable.
    - If A is undecidable then B is undecidable.
    - If B also reduces to A then A and B must be the same problem.
    - If B reduces to C then A reduces to C.
    - If A reduces to C then B reduces to C.
    - If we have an *oracle* program for deciding A, we can use it to solve problem B.
    - If we have an *oracle* program for deciding B, we can use it to solve problem A.
    - Problem A is solvable.
    - Problem B is solvable.
    - Problem A is no harder to solve than problem B.

2. (review from midterm 2) Prove the following reductions, using explicit Python code where needed.

    - HaltsOnEmpty reduces to HaltsOnAllNonempty.
    - HaltsOnAllNonempty reduces to HaltsOnEmpty.
    - HaltsOnEmpty reduces to YesOnString.
    - YesOnString reduces to HaltsOnEmpty.

3. Define what we mean by a (pure) regular expression.

    - What are the basic regular expressions?
    - What are the "regular operations" which can be used to combine given regular expressions to create new expressions?

4. Provide regular expressions for the following languages on the alphabet $\{a, b\}$:

    - All strings that start with three a's and have length 5.

[1] notes/new_cfg_pushdown.html
[2] notes/parsing.html

- All strings that start with three a's and have length at most 5.
- All strings that start with a and end in b, or start with b and end in a.
- All strings with at most 2 b's.
- All strings with at least 2 b's.
- All strings except the string bb.

5. State what the pumping lemma for regular languages says and give a brief sketch of the proof idea.

6. Describe how a pushdown automaton differs from a finite automaton, and what its (non-deterministic) state transitions look like. Explain in simple terms why the PDAs have more computational power than NFAs/DFAs.

7. For the following non-regular languages: prove that they are not regular, build PDAs for them, build context-free grammars for them, and demonstrate the PDA execution as well as a CFG derivation for the provided input strings, *and* a parse tree for the derivation:

   - $\{x^n y^n \mid n \geq 0\}$, $s = x^3y^3$.
   - $\{x^n y^{2n} \mid n \geq 0\}$, $s = x^2y^4$.
   - $\{x^{2n} y^n \mid n \geq 0\}$, $s = x^4y^2$.
   - $\{x^n z^m y^n \mid n, m \geq 0\}$, $s = x^2zy^2$.
   - $\{x^n \mid n \geq 0\} \cup \{x^n y^m \mid n \geq m \geq 0\}$, $s = x^3y^2$.

8. For each of the following, prove or disprove:

   - If L is regular, then L^* is also regular.
   - If L is context-free, then L^* is also context-free.
   - If L_1 and L_2 are regular, then $L_1 \cup L_2$ is also regular.
   - If L_1 and L_2 are context-free, $L_1 \cup L_2$ is also context-free.
   - If L_1 and L_2 are regular, then L_1L_2 is also regular.
   - If L_1 and L_2 are context-free, L_1L_2 is also regular.

9. Describe the two different PDAs corresponding to a given CFG. Follow the derivation of a particular string on the PDAs, showing the evolution of the stack and the process of the input. Recall that one of the PDAs builds a leftmost derivation while the other builds a rightmost derivation.

10. Build the item-set DFAs for the following grammars:

```
S -> epsilon | a | b | aSa | bSb
```

and:

```
E -> a | (A)
A -> E | E A
```

11. Find first and follow sets for the two grammars in the previous question, as well as the polish notation grammar, the basic arithmetic grammar as well as the rewrite of that grammar to avoid left-recursive rules.